UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/409,613	10/01/1999	ARMIN HAROLD CHRISTOFFERSON	R09-99-091	5640

24033 7590 04/23/2003

KONRAD RAYNES VICTOR & MANN, LLP
315 SOUTH BEVERLY DRIVE
SUITE 210
BEVERLY HILLS, CA 90212

EXAMINER
PHAM, HUNG Q

ART UNIT	PAPER NUMBER
2172	12

DATE MAILED: 04/23/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/409,613	CHRISTOFFERSON ET AL.	
	Examiner	Art Unit	
	HUNG Q PHAM	2172	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 March 2003* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Response to Arguments

1. Applicants amended claims 1, 10, and 19, added claims 37-39 in the amendment filed on 03/14/2003. Applicants' arguments have been fully considered but they are not persuasive.

Applicants argued that

> Applicants submit that the cited sections of Harper nowhere teach or suggest the added claim requirement of providing a data structure generated by applying a function to file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate file names used in the file system. Instead, Harper discusses a byte in memory to identify the identifier of elements on a linked list, not file names in a file system as claimed (page 12).
> Moreover, Harper teaches away from applying his hash to file names in a file system. Harper does its hash "by allocating a relatively short block of memory, such as one byte, and by hashing mode identifications to one of the bits of the block of memory." (Harper, col. 4, lines 15) The claimed data structure requires indication of hash values corresponding to file names in a file system and would require far more than one byte of memory allocated because a file system as known in the art must accommodate thousands of file names. See, Application, pg. 6, line 24 to pg. 7, line 14. Thus, Harper teaches away from applying the discussed hash technique for checking duplicates in a linked list to duplicates in a file system because Harper discusses that his "present invention" allocates a short block of memory, such as one byte, as the hash bitmap used to indicate identifiers elements in a linked list (pages 12-13).

Examiner respectfully traverses because of these reasons:

Harper teaches a method, system, and an article of manufacture for utilizing a hash function to speed up the average search for duplicates in a linked list of inode in the UNIX file system (Col. 1, line 5-Col. 2, line15). The structure of link list is shown and disclosed in FIGS. 2-3. As shown in FIG. 8 is the flow chart of the method of checking

whether an element to be added to a linked list is a duplicate of one already on the list. A hash bit map as a data structure is provided and initialized to 0 in block 800. For each element on a linked list, a hash function as in FIG. 4 is applied for hashing the identifier of that element to determine values corresponding to the identifier and map to one bit position of the hash bit map in block 810. The identifier of an element to be added to the linked list is also hashed to one bit location on the bit map in block 820, and that location is checked to determine if the value in the bit location is 1 or 0. If it is 0, no possible duplicate exists and the process returns. If the value of the bit position is 1, a possible duplicate exists. Once a possibility of a duplicate is identified, a comparison of the identifier of the element to be added to the list with the identifier of each element on the list is made to find a duplicate. If a duplicate is found, that fact is reported to the calling process and the process returns (Col. 8, lines 20-35). Harper further discloses the data structure could be a hash table for containing information about the contents of the linked list (Col. 5, lines 29-35). Thus, a data structure such as hash bitmap or hash table is provided, and a hash function is applied to the inode identifiers of the file system to determine values corresponding to the inode identifiers. The hashed value is a position in the hash bitmap or hash table. At the position of the data structure, 0 or 1 as the indicator is utilized for indicating an identifier has been used. Obviously, instead of inode identifiers, file names in the file system could be checked by applying the technique as well because the hash function could be applied to an identifier space, which consist of the set of all legal variable names or file names permitted by the languages (Col. 2, line 57-Col. 3, line 16). In addition, a file system as known in the art

must accommodate thousands of file names could be mapped into a hash table with the size of thousands of indexes.

Applicants argued that

> The Examiner recognized that Harper did not explicitly teach using a hash to identify duplicate file names in a file system, but found that such a modification of Harper would be obvious to apply to file names in a file system. (Third Office Action, pg. 3) Applicants traverse.
> The Manual of Patent Examination Procedure (MPEP) states that the "mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination." MPEP 2143.01, pg. 2100-124 (8 th Ed., Aug. 2001). Here, the Examiner's proposed modification of the cited Harper is improper because the Examiner has not cited any part of Harper or other art that suggests a data structure applying a function, such as a hash function, to file names in a file system to indicate file names used in the file system. There simply is no teaching or suggestion anywhere in the cited art of using a hash for this claimed purpose of determining whether an input file name in a file system is already used in the file system.

In response to applicant's argument that there is no suggestion to combine or modify the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988), and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the Harper method is to determine whether a duplicate of a particular inode exist in a linked list of the UNIX file system by applying a hash function to the inode identifier (Col. 3, lines 48-64). In addition, the hash function could be applied to an identifier space, which consist of the set of all legal variable names or file names permitted by the

languages (Col. 2, line 57-Col. 3, line 16). Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by processing a hash table as a data structure to determine a possible duplication of file name, and by using the step of processing, a duplication of file name in the file system could be identified quickly by scanning the hash table.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 1-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Harper [USP 5,765,165].

Regarding to claims 1, 10, and 19, Harper teaches a method, system, and an article of manufacture for utilizing a hash function to speed up the average search for duplicates of file in a UNIX file system (Col. 1, lines 5-49). The Harper method could be used either to detect whether duplicates exist on a linked list or to detect whether an element to be added to a linked list is a duplicate of one which already exists on the linked list (Col. 4, lines 21-29). As shown in FIG. 1-3C, the relationship between the data blocks, the inode, and a directory that references the file is illustrated (Col. 1, line 13-16). As shown in FIG. 8, in order to check whether an element such as a file to be added to a linked list is a duplicate of one already on the list, a hash function to hash the identifier of the added element to one bit position of the hash bit map at step 820 as the step of ***applying a function to map the input file name to a value***. Harper does not explicitly discloses the step of ***providing a data structure generated by applying a function to file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate file names used in the file system; processing a data structure to determine whether there is a preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name***. However, as shown in FIG. 8 is the flow chart of the method of checking whether an element to be added to a linked list is a duplicate of one

already on the list. A hash bit map as a data structure is provided and initialized to 0 in block 800. For each element on a linked list, a hash function as in FIG. 4 is applied for hashing the identifier of that element to determine values corresponding to the identifier and map to one bit position of the hash bit map in block 810. The identifier of an element to be added to the linked list is also hashed to one bit location on the bit map in block 820, and that location is checked to determine if the value in the bit location is 1 or 0. If it is 0, no possible duplicate exists and the process returns. If the value of the bit position is 1, a possible duplicate exists. Once a possibility of a duplicate is identified, a comparison of the identifier of the element to be added to the list with the identifier of each element on the list is made to find a duplicate. If a duplicate is found, that fact is reported to the calling process and the process returns (Col. 8, lines 20-35). Harper further discloses the data structure could be a hash table for containing information about the contents of the linked list (Col. 5, lines 29-35). Thus, a data structure such as hash bitmap or hash table is provided, and a hash function is applied to the inode identifiers of the file system to determine values corresponding to the inode identifiers. The hashed value is a position in the hash bitmap or hash table. At the position of the data structure, 0 or 1 as the indicator is utilized for indicating an identifier has been used. This technique indicates the steps of ***providing a data structure generated by applying a function to*** inode identifiers ***in a file system to determine values corresponding to the*** inode identifiers, ***wherein the data structure indicates those values corresponding to the*** inode identifiers ***to indicate*** inode identifiers ***used in the file system; processing a data structure to determine whether there is a preexisting*** inode ***in the file system having a name***

that maps, according to the function, to the same value to which the input inode identifier ***maps, wherein two*** inodes ***that map to a same value according to the function are capable of having a same name***. Harper further discloses the hash function could be applied to an identifier space, which consists of the set of all legal variable names or file names permitted by the languages (Col. 2, line 57-Col. 3, line 16). Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by processing a hash bit map as a data structure to determine a possible duplication of file name, and by using the step of processing, a duplication of file name in the file system could be identified quickly by scanning the hash bit map.

Regarding to claims 2, 11 and 20, Harper teaches all the claimed subject matters as discussed in claims 1, 10 and 19, Harper further discloses ***the mapped-to values require fewer bits of storage than the file names*** (Col. 4, lines 29-34).

Regarding to claims 3, 12 and 21, Harper teaches all the claimed subject matters as discussed in claims 1, 10 and 19, Harper further discloses ***the function is a hash function that maps the input file name to an integer value, and wherein the data structure includes an entry for each possible integer value capable of being generated from the hash function*** (Col. 2, line 57-Col. 3, line 27 and Col. 4, lines 29-34).

Regarding to claims 4, 13 and 22, Harper teaches all the claimed subject matters as discussed in claims 3, 10, 21, Harper further discloses the step of ***determining whether the entry for the integer value to which the input file name maps indicates the presence of one preexisting file mapping to the same integer value as the input file name*** (Col. 8, lines 25-35).

Regarding to claims 5, 14 and 23, Harper teaches all the claimed subject matters as discussed in claims 4, 13 and 22, Harper further discloses ***the data structure is a one-dimensional array and wherein each entry is capable of having one of two values, further comprising setting the entry to a first value if there is one preexisting file name in the file system that maps to the integer value for the entry, and wherein determining whether there is one preexisting file comprises determining whether the entry for the integer value to which the input file name maps has the first value*** (Col. 5, lines 29-35 and Col. 8, lines 20-35).

Regarding to claims 6, 15 and 24, Harper teaches all the claimed subject matters as discussed in claims 1, 10, 19, Harper further discloses the steps of ***applying the function to each file name in the file system to map each file name to one value; and indicating in the data structure, for each file name, that there is one preexisting file for the value to which the file name maps*** (Col. 8, lines 20-28).

Regarding to claims 7, 16 and 25, Harper teaches all the claimed subject matters as discussed in claims 6, 15 and 24, Harper further discloses the step of ***scanning each***

file in the file system to determine if there is at least one preexisting file having the same name as the input file name if there is one preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps (Col. 8, lines 29-35).

Regarding to claims 8, 17 and 26, Harper teaches all the claimed subject matters as discussed in claims 7, 16 and 25, Harper does not explicitly disclose the step of ***adding the input file as a new file to the file system if no preexisting file in the file system has the same name as the input file name; and rejecting the access request if there is a preexisting file in the file system having the same name***. However, as shown in FIG. 8, when an element such as a file is added to the file system, the element identifier is checked for a possible duplicate and if there is no duplicate, the process return at step 850, otherwise the fact is reported to the calling process. Thus, the calling process is the process for adding a file to the file system and obviously, the Harper process of adding a file will ***add the input file as a new file to the file system if no preexisting file in the file system has the same name as the input file name; and rejecting the access request if there is a preexisting file in the file system having the same name*** by displaying a message as in most of conventional file system such as Window 95. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by including the step of adding the input file and rejecting the input file in order to add a file to the file system.

Regarding to claims 9, 18 and 27, Harper teaches all the claimed subject matters as discussed in claims 7, 16 and 25, but fails to disclose the step of ***updating a preexisting file in the file system having the same name as the input file with the data in the input file if there is such a preexisting file; and rejecting the access request if there is no preexisting file in the file system having the same name as the input file name***. However, as shown in FIG. 8, when an element such as a file is added to the file system, the element identifier is checked for a possible duplicate and if there is no duplicate, the process return at step 850, otherwise the fact is reported to the calling process. In order to update a preexisting file, the Harper process could be modified by return to the calling process if a duplicate occurs and report to the calling process if there is no match. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by including the step of updating if there is a preexisting file and rejecting if there is no preexisting file in order to update a file in a file system.

Regarding to claims 28, 31 and 34, Harper teaches all the claimed subject matters as discussed in claims 1, 10 and 19, Harper further discloses the step of ***searching the file system for one preexisting file having the same name as the input file name if the data structure indicates that one preexisting file has a name that maps, according to the function, to the same value to which the input file maps; and performing an operation if the file system includes one preexisting file having the same name as the input file*** (Col. 8, lines 25-35).

Regarding to claims 29, 32 and 35, Harper teaches all the claimed subject matters as discussed in claims 28, 31 and 34, Harper does not explicitly disclose the step of ***applying update data to the preexisting file having the same name as the input file if the file system includes one preexisting file having the same name as the input file***. However, as shown in FIG. 1, Harper teaches that the file name "proposal" is associated with an inode number pointing to the inode containing information about the file "proposal." The inode number "248231" points to inode 24823 which contains information about the file "proposal" such as when the file was created, and when it was last modified. The inode points to data blocks. The data blocks contain the actual characters in the file such as the textual contents of the "proposal" after which the file was named (Col. 1, lines 40-49). Thus, if the calling process is an updating process, and there is a preexisting file having the same name as the input file after comparing the identifier with identifier of each element on the list (Col. 8, lines 25-35), obviously, the updated data such as date, and the actual data of the file will be applied to the preexisting file. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by including the step of applying update data to the preexisting file in order to update a file in a file system.

Regarding to claims 30, 33 and 36, Harper teaches all the claimed subject matters as discussed in claims 28, 31, 34, Harper does not explicitly disclose the steps of ***returning an error if the file system includes one preexisting file having the same name as***

the input file; and adding the input file to the file system if the file system does not include one preexisting file having the same name as the input file. However, as shown in FIG. 8, when an element such as a file is added to the file system, the element identifier is checked for a possible duplicate and if there is no duplicate, the process return at step 850, otherwise the fact is reported to the calling process. Thus, the calling process is the process for adding a file to the file system. And obviously, the step of adding the file to the file system will be executed if there is no duplicate, otherwise, the fact is reported as the step of returning an error. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by using the step of returning an error or adding the input file name in order to add a file to a file system.

4. Claims 37-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Harper [USP 5,765,165] in view of Williams [USP 5,990,810].

Regarding to claims 37-39, Harper teaches all the claimed subject matters as discussed in claim 1, 10, and 10, but fails to disclose the function comprises ***a wide hash function to produce a large number of possible hash values to minimize the likelihood that the application of the hash function to file names in the file system would have a same hash value.*** Williams teaches ***a wide hash function to produce a large number of possible hash values to minimize the likelihood that the application of the hash function to file names in the file system would have a same hash value*** (Williams, Col. 11, lines 25-45). Therefore, it would

have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method, system, and an article of manufacture by using a wide hash function in order to avoid the collision when applying the hash function.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hung Pham whose telephone number is 703-605 4242. The examiner can normally be reached on Monday-Friday, 7:00 Am - 3:30 Pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, VU, KIM YEN can be reached on 703-305 4393. The fax phone numbers

for the organization where this application or proceeding is assigned are 703-746 7239 for regular communications and 703-746 7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305 3900.

Examiner: Hung Pham
April 14, 2003

JEAN M. CORRIELUS
PRIMARY EXAMINER